**Weekly Assessment – DevOps**

**Task: Setting up Jenkins for Continuous Integration/Continuous Deployment (CI/CD)**

Description:

Configure Jenkins to automate the build, test, and deployment processes for your application.

Jenkins Installation and Configuration:

Test Case 1: Verify that Jenkins is successfully installed on the designated server.

Expected Outcome: Jenkins should be installed without errors, and the web interface should be

accessible.

2. CI Pipeline Execution:

Test Case 2: Trigger a sample commit in the version control system and ensure that the CI

pipeline is automatically executed.

Expected Outcome: The CI pipeline should fetch the latest code, compile it, run automated

tests, and produce deployable artifacts.

3. CD Pipeline and Deployment:

Test Case 3: Manually trigger the CD pipeline and confirm that the application is deployed to a

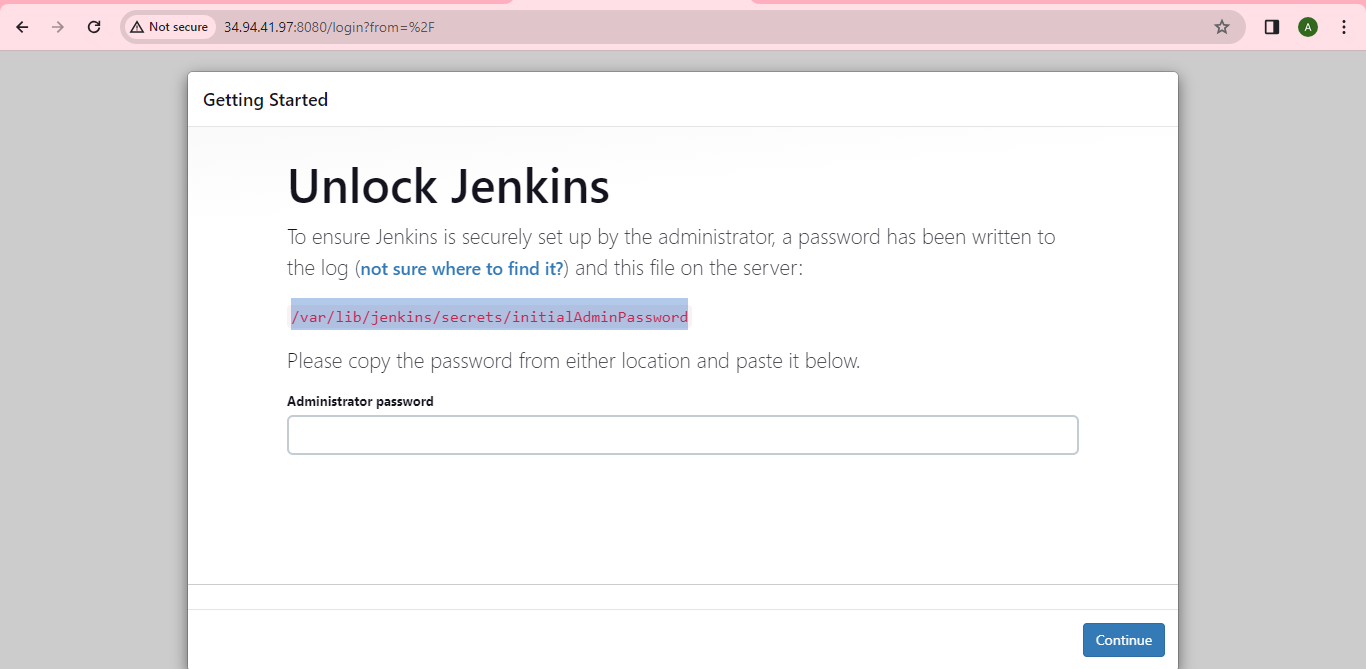
staging environment.

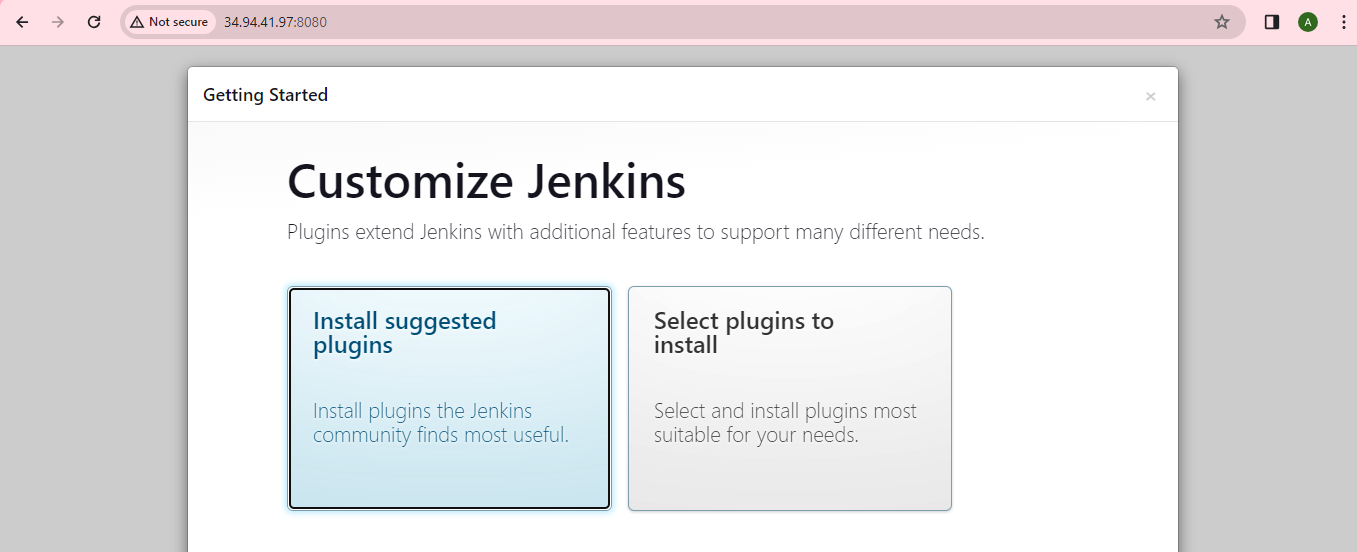
Expected Outcome: The CD pipeline should deploy the application successfully, and the staging

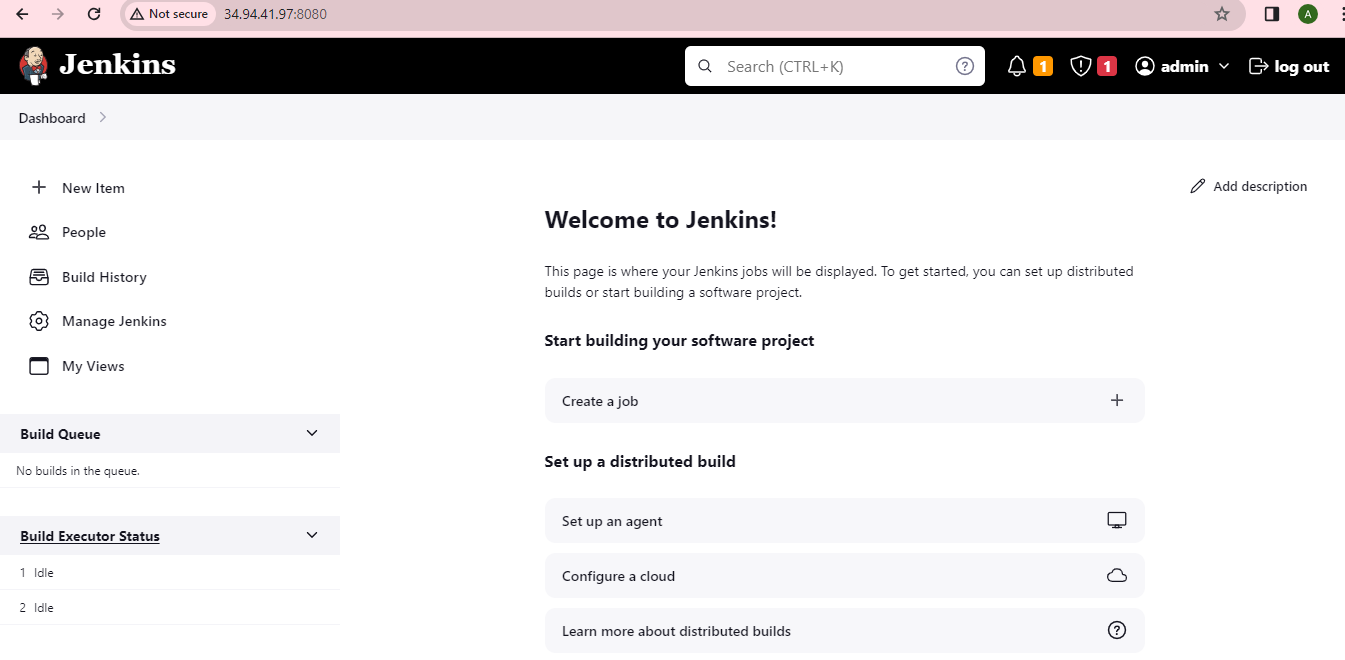
environment should reflect the latest changes.

**Test Case 1: Verify that Jenkins is successfully installed on the designated server.**

**Expected Outcome: Jenkins should be installed without errors, and the web interface should be accessible.**

****

****

****

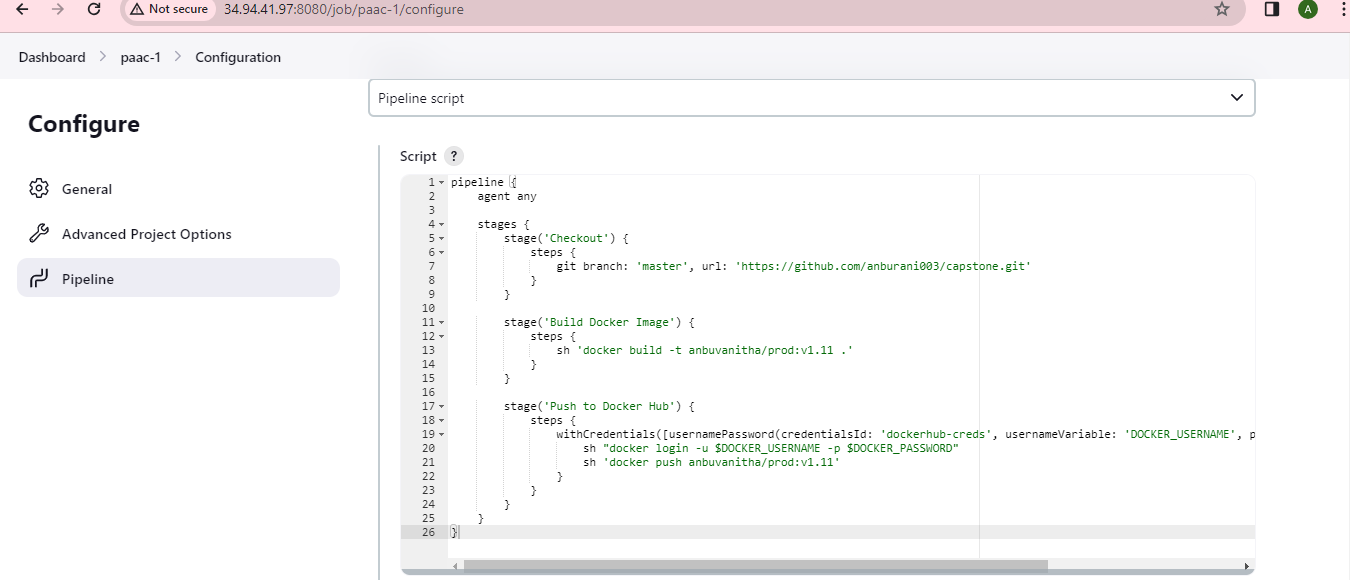
**2. CI Pipeline Execution:**

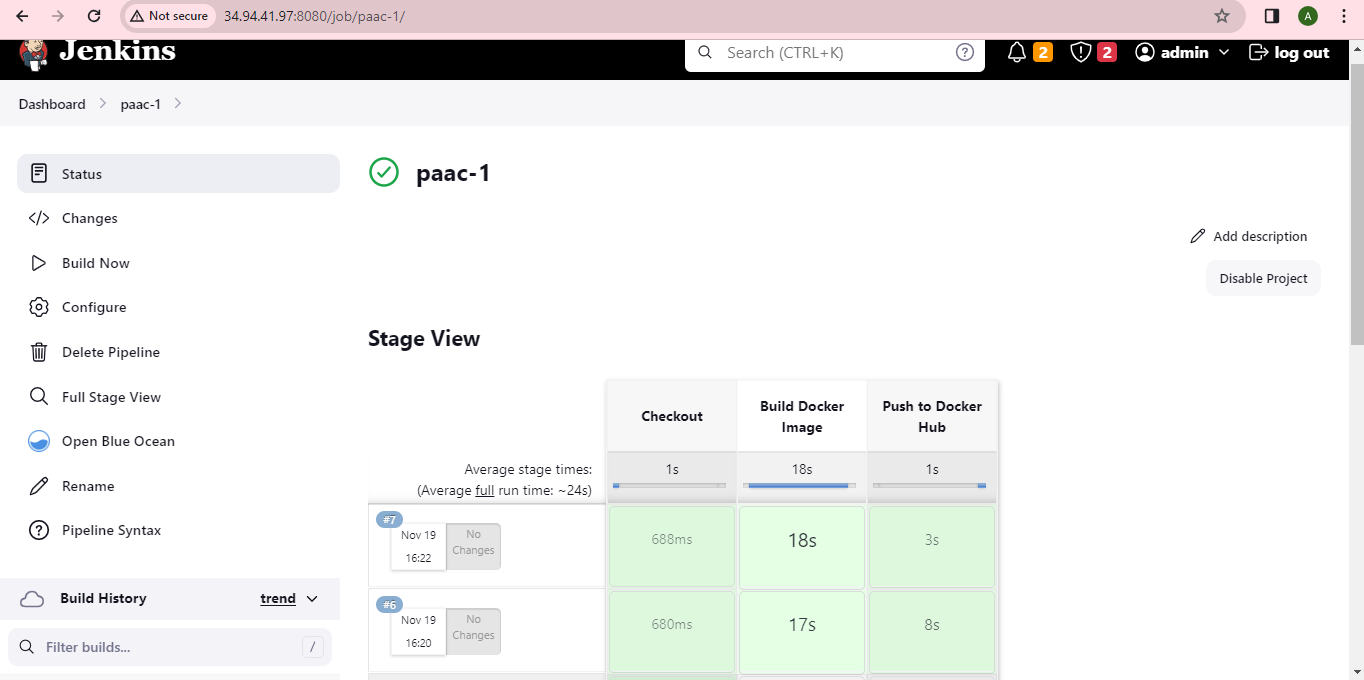
**Test Case 2: Trigger a sample commit in the version control system and ensure that the CI**

**pipeline is automatically executed.**

Expected Outcome: The CI pipeline should fetch the latest code, compile it, run automated

tests, and produce deployable artifacts.





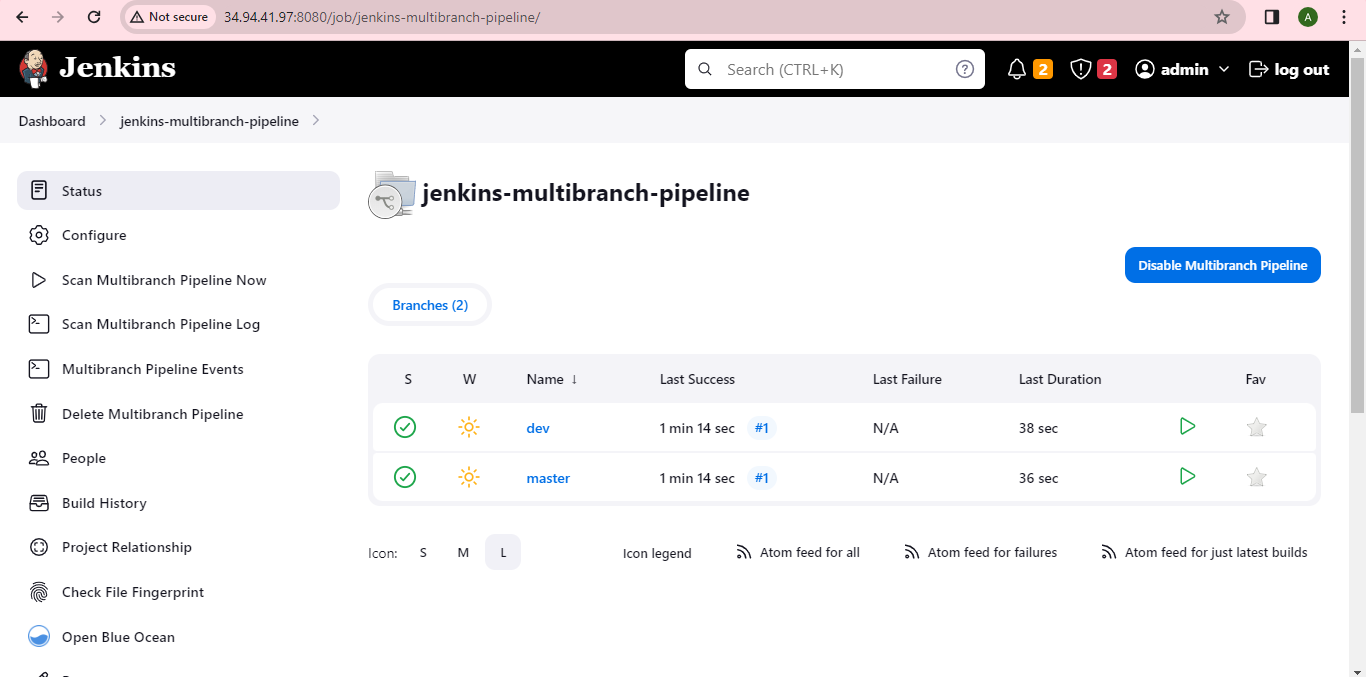
**3. CD Pipeline and Deployment:**

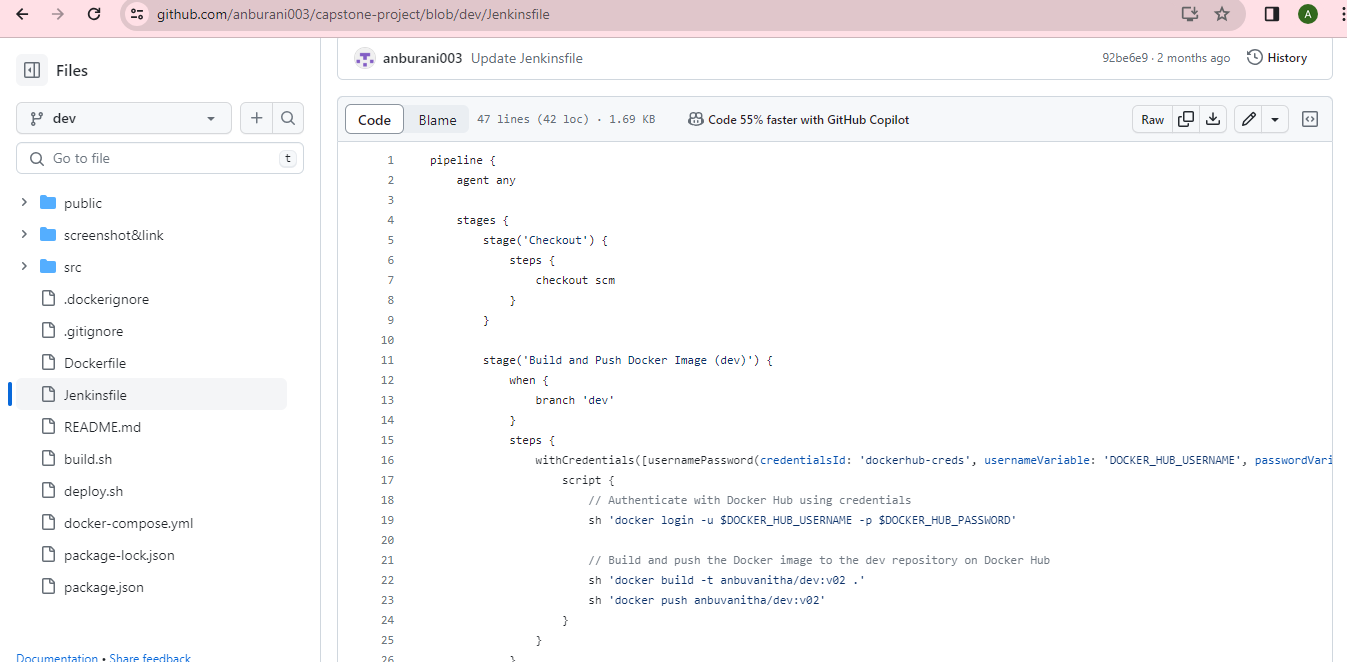
Test Case 3: Manually trigger the CD pipeline and confirm that the application is deployed to a

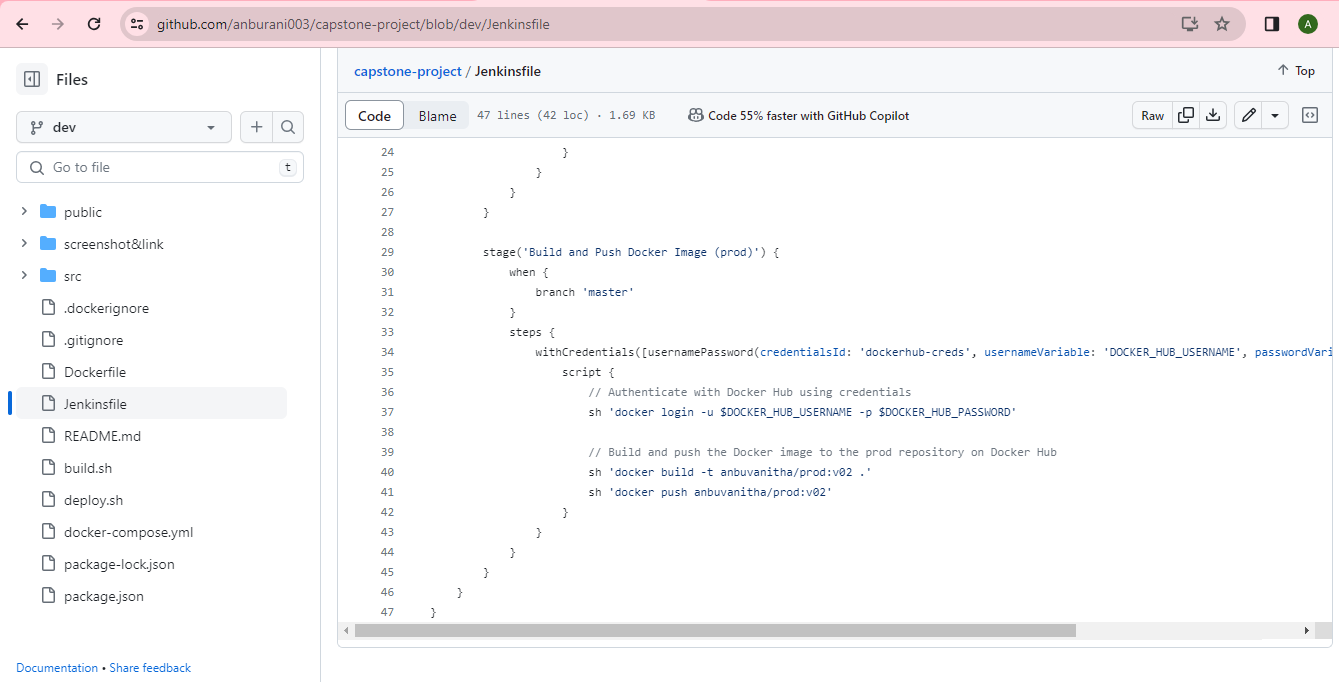
staging environment.

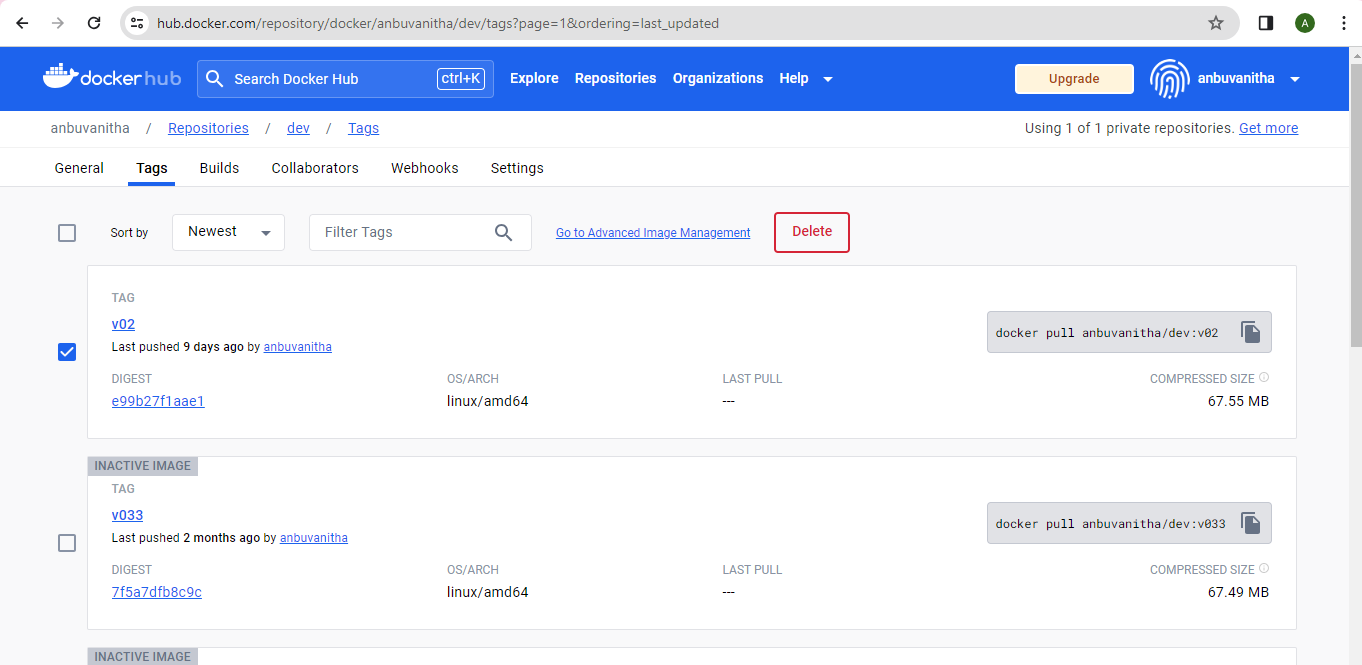
Expected Outcome: The CD pipeline should deploy the application successfully, and the staging

environment should reflect the latest changes

****

****

****

****

**Jenkins Installation and Configuration:**

**Step 1: Install Jenkins**

* SSH into your designated server.
* Update the package list:

sudo apt update

* Install Jenkins:

sudo apt install jenkins

* Start Jenkins service:

sudo systemctl start jenkins

* Enable Jenkins to start on boot:

sudo systemctl enable jenkins

* Open a web browser and navigate to http://your\_server\_ip:8080. Retrieve the initial password from:

sudo cat /var/lib/jenkins/secrets/initialAdminPassword

Follow the instructions to complete the Jenkins setup.

Test Case 1: Verify Jenkins Installation

Expected Outcome: Jenkins should be installed without errors, and the web interface should be accessible.

**CI Pipeline Execution:**

**Step 2: Set Up CI Pipeline**

* Install necessary plugins in Jenkins:
* Git Plugin
* Pipeline Plugin
* Create a new Jenkins pipeline job:
* Choose "Pipeline" as the job type.
* Configure your version control system (e.g., Git) credentials.
* In the pipeline script, define the stages:

pipeline {

agent any

stages {

stage('Checkout') {

steps {

checkout scm

}

}

stage('Build') {

steps {

// Add build commands here

}

}

stage('Test') {

steps {

// Add test commands here

}

}

stage('Deploy') {

steps {

// Add deployment commands here

}

}

}

}

Save and run the pipeline manually to ensure it works.

Test Case 2: Trigger CI Pipeline

Expected Outcome: After a sample commit, the CI pipeline should fetch the latest code, compile it, run automated tests, and produce deployable artifacts.

**CD Pipeline and Deployment:**

**Step 3: Set Up CD Pipeline**

* Install necessary plugins in Jenkins:
* Deploy to Container Plugin (if deploying to containers) Pipeline Plugin
* Extend your Jenkinsfile to include CD stages after the successful completion of CI stages.

stages {

// ... CI stages

stage('Deploy to Staging') {

steps {

// Add deployment commands to staging environment

}

}

stage('Deploy to Production') {

when {

// Define conditions for deploying to production (e.g., manual approval)

}

steps {

// Add deployment commands to production environment

}

}

}

* Save and run the pipeline manually to ensure it works.

Test Case 3: Trigger CD Pipeline

* Set up the CD pipeline to be manually triggered.
* Manually trigger the CD pipeline.
* Run the CD pipeline manually to deploy to the staging environment.
* Confirm that the application is deployed successfully to the staging environment.
* Verify that the staging environment reflects the latest changes.